FIG. **4** is an example embodiment of an extended configuration of an example embodiment of the present invention;

FIG. **5** illustrates the communication paths between the VLX and LNX structures in accordance with an example embodiment of the present invention;

FIG. **6** illustrates the functional layering of an example MXE in accordance with the present invention;

FIG. **7** is an example cluster in accordance with the present invention;

FIG. **8** is a simplified block diagram illustrating the communications links between the front-end and back-end of an example embodiment of the present invention;

FIG. **9** is an example LNX software module containing data structures in accordance with an example embodiment of the present invention;

FIG. **10** is an example set of data structures for the VLX software module in accordance with an example embodiment of the present invention,

FIG. **11** illustrates the data structures maintained for each call which include an array of call nodes containing channel-specific call information;

FIG. **12** illustrates the front-end machine lookup-tables and the different sources from which lookup-table data may be loaded in accordance with an example embodiment of the present invention;

FIG. **13** is an example mailbox table from a front-end lookup-table in accordance with an example embodiment of the present invention; and

FIG. **14** is a schematic diagram of a front-end and LNX combination for an example multiple clustering embodiment of the present invention.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENTS

The present system is based on a concept that allows for cost-effectively increasing the performance and capacity of a single messaging node by creating a cluster of nodes that appear, from the network's point of view, to be a single high-capacity system providing single number access. Of equal importance, however, is that each individual node of the cluster operates as a stand-alone system. This allows a single software image to exist for all machines.

The present system is built upon the concept of a "virtual" channel, and the idea of a "front-end" node that can, using these virtual channels, switch a call to any one of several back-end machines.

FIG. **1** illustrates an example embodiment of such a system. In essence, this example cluster **10** includes four MXEs (back-end **1**, back-end **2**, back-end **3** and front-end) and a single node used to provide the signal interfacing to the telecom environment (which may be in the current environment the "LNX" box and software described previously). Each of the back-end MXEs is connected to the LNX via real MMI channels **11**. The LNX in turn, is connected to the communication network via network channels **14**. Without the clustering arrangement, a single MXE would ordinarily receive messages from the LNX referring to real network channels. But, in the present example, the capacity of the system is increased by providing three back-end systems (instead of just one), which are coordinated by the front-end system.

Each of the back-end MXEs communicate with the front-end via virtual channel signaling links **12** between each of the respective back-ends and the single front-end. The virtual channel signaling links between the back-end and the front-end carry information about the virtual channels (and carry subscriber lists, as discussed in more detail below). Although the back-ends are allocating the MMI channels, the front-end will allocate the virtual channels identified in the virtual channel signaling links **12**. The front-end then communicates with the LNX controller via the real network signaling link **13**, which carries information about real network channels (and carries an LNX software image). In this way, the front-end provides a translation from the virtual channels identified via the virtual channel links **12** to the real network channels and instructs the LNX to cross-connect the real MMI channels with the network channels based on this translation and instructions from the back-end.

Each back-end MXE will have associated with it a list of subscribers that reside on the respective back-end machine. This list of subscribers must be communicated to the front-end and is therefore transferred over the signaling links **12** to the front-end machine where they are loaded into a lookup-table. This lookup-table allows the front-end machine to translate the virtual channels to the real network channels for the respective subscribers. The lists of subscribers from the back-end machines are transferred in the same way that a software image is transferred to the LNX. That is, in all respects, the back-ends operate just as though they were talking to the LNX in the normal fashion, thus preserving the transparency of the cluster.

Each of the signaling links **12** between the back-ends and the front-end will have corresponding groups of the virtual channels discussed previously. These virtual channels don't really exist to the extent that they have physical circuits. Instead, these channels are given to the back-end machine to allow the back-end machine to think such channels exist between the LNX and the network via the real network channels **14**. This allows the back-end machines to preserve a single image of the software, just as though the back-end machine was used as a stand-alone MXE. As a result, to create the present cluster, no new development is necessary for any of the back-end machines, because they continue to operate just as though they were in a stand-alone mode. In the end, it is the front-end machine which translates the virtual channels to the real network channels and instructs the LNX via the real network signaling link **13** accordingly. At this point virtual channels communicated on link **12** have been translated to real network channels **14**. The LNX then cross connects the MMI channels **11** with the real network channels **14**.

1. TELEPHONY CHANNEL CONFIGURATION

Two databases employed by the example embodiment of the present invention are the Telephone Controller (TCR) database, and the Telephony Channel (TCH) database. An entry in the TCH database can be viewed as the intelligence of a port that a call would originate or terminate on, meaning each TCH entry can include a storage location containing the definition for the signaling used, instructions on how the call should be processed, etc. An entry in the TCR database would be the parent controller of a group of telephony channels, meaning the processor that executed the software required by the telephony channel.

2. MXE MODULARITY AND SYSTEM OVERVIEW

The base component employed in both the front-end and back-ends of the clusters in the example embodiments is the MXE (**18** in FIG. **2**). One could view the MXE from a very high level as consisting of several layers of software. The lowest layer, for example, could be a UNIX operating